

Random sampling: Billiard Walk algorithm

Elena Gryazina^a, Boris Polyak^b

^a*Institute for Control Sciences RAS, Moscow, Russia*

^b*Institute for Control Sciences RAS and Laboratory of Structural Methods of Data Analysis in Predictive Modeling in Moscow Institute of Physics and Technology, Moscow, Russia*

Abstract

Hit-and-Run is known to be one of the best random sampling algorithms, its mixing time is polynomial in dimension. Nevertheless, in practice the number of steps required to achieve uniformly distributed samples is rather high. We propose new random walk algorithm based on billiard trajectories. Numerical experiments demonstrate much faster convergence to uniform distribution.

Keywords: Sampling, Monte-Carlo, Hit-and-Run, Billiards

1. Introduction

Generating points uniformly distributed in an arbitrary bounded region $Q \subset \mathbb{R}^n$ (sampling) finds applications in many computational problems [1, 2].

Straightforward sampling techniques are usually based on one of three approaches: rejection, transformation and composition. In rejection approach the region of interest Q is enclosed within the region with available uniform sampler B (usually a box or a ball). At the next step non belonging to Q samples are rejected. Suppose Q is an unit ball while bounded region B is a box $[-1, 1]^n$. Then for $n = 2k$ we obtain the ratio of volumes of the box and the ball $q = \frac{\text{Vol}(Q)}{\text{Vol}(B)} = \frac{\pi^k}{k!2^k}$, thus $q \sim 10^{-8}$ for $n = 20$ and we should generate $\sim 10^8$ samples to have just a few in Q . For polytopes this ratio can be much smaller. The other way to exploit pseudo-random number generator for simple region B is to map B onto Q via smooth deterministic function with constant Jacobian. For instance, to obtain uniform samples in $Q = \{x : x^T A x \leq 1\}$, A being positive definite matrix, it suffices to take y uniform in the unit ball $\|y\|_2 \leq 1$ and transform them as $x = A^{-1/2}y$. Unfortunately, such a transformation exists just for a limited class of regions.

In composition approach we partition Q for finite number of sets that can be efficiently sampled. Apart from narrow class of regions with available partition, Q is partitioned into finite union of simplices and the number of simplices makes the procedure computationally hard.

Other sampling procedures use modern versions of Monte Carlo technique, based on Markov Chain Monte Carlo (MCMC) approach [3, 4]. For instance, recent efficient algorithms for volume computation based on random walks can be found in [5, 6]. One of the most famous and effective algorithms of MCMC type is Hit-and-Run (HR) that was originally proposed by Turchin [7] and independently by Smith [8]. The examples of Hit-and-Run applied to various control and optimization problems are provided in [9, 10, 11]. Unfortunately, even for simple “bad” sets, such as level sets of ill-posed functions, HR techniques fail or at least are computationally inefficient. A variety of applications and drawbacks of existing techniques propose much room for improvement new sampling algorithms. For instance, there were attempts to exploit the approach, developed for interior-point methods of convex optimization [12], and to combine it with MCMC algorithms. As a result Barrier Monte Carlo method [13] generates random points that are preferable in comparison with standard Hit-and-Run. But the complexity of each iteration in general is high enough (the calculation of $(\nabla^2 F(x))^{-1/2}$, where $F(x)$ is a barrier function of the set, is needed). Moreover such approach can not accelerate convergence for sets like simplices.

In this paper we propose the new random walk algorithm, which is motivated by physical phenomena of a gas diffusing in a vessel. A particle of gas moves with constant speed until it meets a boundary of the vessel, then it reflects (the angle of incidence equals the angle of reflection) and so on. When the particle hits another one, its direction and speed changes. In our simplified model we assume that direction changes randomly while speed remains the same. Thus our model combines ideas of Hit-and-Run technique with use of billiard trajectories. There exist a vast literature on mathematical billiards, and many useful facts can be extracted from there [14, 15, 16, 17, 18]. Traditional theory addresses the behavior of one particular billiard trajectory in different billiard tables, their ergodic properties and the conditions for existence of periodic orbits. We extend billiard trajectories with random change of directions, this introduction of randomness enriches their ergodic properties.

The paper is organized as follows. In Section 2 we present novel sampling

algorithm and prove that it produces asymptotically uniformly distributed samples in Q . Section 3 is dedicated to discussion of some properties of the Billiard Walk, implementation issues are discussed as well. Simulation of BW for particular test domains is presented in Section 4. Much attention is devoted to ability of BW to get out of the corner in comparison with HR. Here we consider just the most demonstrative types of geometry. In Section 5 we briefly discuss possible applications of the algorithm.

2. Algorithm

Suppose there is a bounded closed connected set $Q \subset \mathbb{R}^n$ and a point $x^0 \in Q$. Our aim is to generate asymptotically uniform samples $x^i \in Q$, $i = 1, \dots, N$.

The brief description of Hit-and-Run algorithm is as follows. At every step HR generates a random direction uniformly over the unit sphere and chooses next point uniformly from the segment of the line in given direction in Q .

New algorithm Billiard Walk (BW) generates a random direction uniformly as Hit-and-Run. But the next point is chosen as the end of the billiard trajectory of length ℓ . This length is chosen randomly: we assume that probability of collision with another particle is proportional to δt for small time instances δt , this validates the formula for ℓ in algorithm below. The scheme of the method is given in Fig. 1 while the precise routine is as follows.

Algorithm of Billiard Walk.

1. Starting point $x^0 \in \text{Int } Q$ is given; $i = 0$, $x = x^0$.
2. Generate the length of the trajectory $\ell = -\tau \log \xi$, ξ being uniform random in $[0, 1]$, τ is a specified parameter of the algorithm.
3. Pick random direction $d \in \mathbb{R}^n$ uniformly distributed on the unit sphere (i.e., $d^i = \xi / \|\xi\|$, $\xi = \text{randn}(n, 1)$ – the n -dimensional vector with normally distributed components). Construct billiard trajectory starting at x^i and with initial direction $d = d^i$. When the trajectory meets a boundary with internal normal s , $\|s\| = 1$, the direction is changed as

$$d \rightarrow d - 2(d, s)s.$$

4. Calculate the end of the trajectory of length ℓ . If the point with nonsmooth boundary is met or the number of reflections exceeds $10n$ go to step 3.

5. $i = i + 1$, take the end point as x^{i+1} and go to step 2.

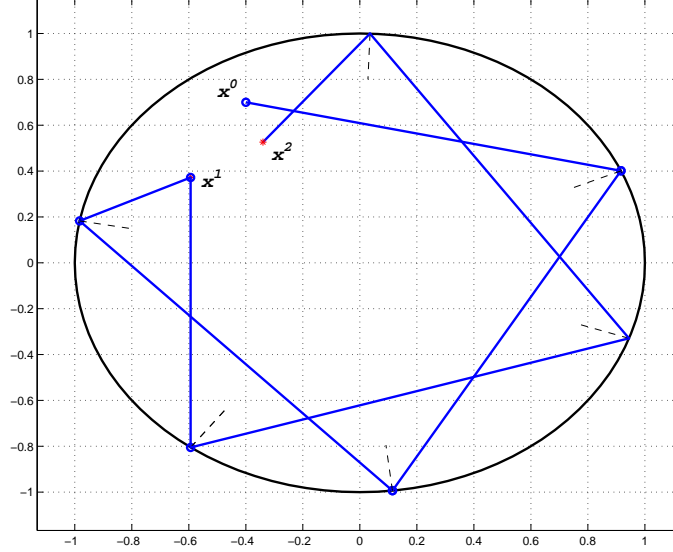


Figure 1: Billiard Walk.

Theorem 1. *Suppose Q is connected, bounded and open (or a closure of such set) set, the boundary of Q is piecewise smooth. Then the distribution of points x^i sampled by BW algorithm tends to uniform on Q .*

Proof. First, the algorithm is well defined: with probability one x^{i+1} will be found for arbitrary $x^i \in \text{Int } Q$ (i.e. billiard trajectory with arbitrary initial conditions can be extended for arbitrary finite length). Note that we restrict the number of reflections to avoid situations when the trajectory comes (in limit) to a point with no normal with nonzero probability and for fixed ℓ the length of billiard trajectory remains $< \ell$ after any number of reflections.

All the restrictions for Q are important. Connectedness guarantees that stating from any point we can reach any other point of Q . Boundedness is necessary to define uniform distribution on Q and to avoid trajectories going to infinity. Openness allows us to connect any two points with a tube of nonzero measure. Thus, there exists piecewise linear trajectory connecting arbitrary points.

In view of Theorem 1 in [8] it suffices to prove that $p(y|x) > 0$ for all $x, y \in \text{Int}Q$ and $p(y|x) = p(x|y)$, where $p(y|x)$ is probability density of transition from x to y . Inequality $p(y|x) > 0$ is guaranteed because all the directions are possible, Q is connected and open, and probability of any length ℓ is positive. Equality $p(y|x) = p(x|y)$ (reversibility) holds due to reflection law: the angle of incidence equals the angle of reflection. Thus both conditions are satisfied and the distribution of points x^i sampled by BW algorithm tends to uniform on Q . \square

3. Discussion

We start discussion from some implementation issues.

3.1. Choice of τ .

To run the algorithm parameter τ need to be specified. The value of τ strongly influences the behavior of the method. When τ is small enough BW becomes slower than HR, it behaves as a ball walk with radius τ . Empirical observations show that fast convergence to uniform distribution is achieved for $\tau \approx \text{diam}Q$.

3.2. Preliminary transformation of Q .

If Q is “ill-shaped”, sometimes it can be improved with its linear transformation. For instance, if Q is a box $Q = \{x : |x_i| \leq a_i\}$ with a_i having large data scattering simple scaling transforms Q into a cube. Similar scaling can be done to transform an ellipsoid into a ball. In general case the following scaling can be helpful. Suppose Q has a barrier $F(x)$ [12]; for instance, for the polytope $Q = \{x : (a_i, x) \leq b_i\}$ this barrier is $F(x) = -\sum_i \log(b_i - (a_i, x))$.

Then it is an easy task to find x^* — an approximate minimum of $F(x)$. *Dikin ellipsoid* $E = \{x : (H(x - x^*), (x - x^*)) \leq 1\}$, $H = \nabla^2 F(x^*)$ lies in Q and is a good approximation of the polytope Q . Hence we can calculate linear mapping $T = H^{-1/2}$; generating directions $d' = Td$, where d is uniformly distributed on the unit sphere, we can strongly accelerate the convergence. However sometimes none of transformations can improve the shape of the set, a simplex is known to be the worst-case example.

3.3. Boundary oracle and normals

Both HR and BW algorithms require the calculation of intersection of a straight line (defined by point x^k and direction d of the trajectory) with the set Q . We call this segment *Boundary Oracle* (BO) that is the segment $[\underline{t}, \bar{t}]$, where

$$\underline{t} = \max_{t < 0} \{t : x^k + td \in \partial Q\}, \quad \bar{t} = \min_{t > 0} \{t : x^k + td \in \partial Q\}$$

(we suppose that Q is convex, otherwise the points of first intersection of straight line and boundary of Q are taken). In most applications finding BO is not a problem. For instance, if Q is a polytope

$$Q = \{x \in R^n : (a^i, x) \leq b_i, \ i = 1, \dots, m\}$$

then

$$t_i = \frac{b_i - (a^i, x^k)}{(a^i, d)}, \quad i = 1, \dots, m,$$

$$\underline{t} = \max_{t < 0} t, \quad \bar{t} = \min_{t > 0} t.$$

Numerous examples of BO for other sets Q (for instance, defined by *Linear Matrix Inequalities*) can be found in [9, 10, 11].

BW walks requires also calculation of normals s in boundary points. In most applications it is not hard, for instance, for a polytope $s = a_i$, where i is the index, for which maximum or minimum in above formulas is achieved.

3.4. Comparison of HR and BW

Our goal in test examples below is to compare HR and BW. We use several tools for this purpose. Sometimes theoretical considerations can help to compare the number of iterations to quit from the corner. It is well known, that HR can require too many iterations to get out of the corner, see for instance estimates in [19]. We will show that estimates for BW are much more optimistic for many particular examples. On the other hand, we use simulation for comparison as well. We exploit different tools to demonstrate that one sampling is closer to uniform than another. Sometimes graphical figures in 2D plane are quite evident. In other cases we demonstrate strong serial correlation in samples (for instance, distance to initial point increases too slowly). Finally, we use parametric partition of Q and compare the

number of empirical points compared with theoretical number for uniform distribution.

To make final conclusions on comparison of both methods we should have in mind the following. Of course, computationally BW is harder than HR. It requires more BO calculations, each reflection at the boundary also requires extra calculations, and it is not obvious how to estimate the complexity of each iteration of BW. Nevertheless acceleration of convergence to uniform distribution often makes BW preferable if compared with HR.

4. Test sets and simulation

Some sets below are unbounded; they are considered just to analyze the behavior at a corner. We say that a trajectory quits the corner if it goes to infinity.

4.1. Angle at a plane

Let $Q \subset \mathbb{R}^2$ be an angle equal $\alpha < \pi$. Then billiard trajectory independently of initial point and initial direction quits Q after no more than $N^* = \lfloor \pi/\alpha \rfloor$ reflections, here $\lfloor a \rfloor$ stands for smallest integer $\geq a$. It can be proved this way: if we reflect our angle N times around its side, billiard trajectory will be a straight line. It can not intersect any straight line twice.

For HR we quit Q with probability $1 - (1 - \alpha/\pi)^N$ after N iterations. For $N = N^*$ large we quit Q with probability $1 - 1/e = 0.63$ after N^* iterations, while for BW we do it w.p.1.

It is of interest to estimate an average number of reflections (over random initial directions). Consider the triangle $Q = \{x \in \mathbb{R}^2 : |x_1| \leq \tan \frac{\alpha}{2}, x_2 \leq 1\}$ with angle α . Let BW trajectories start at $[0; 0.1]$ and calculate the number of reflections until the trajectory reaches the line $x_2 = 1$. For $\alpha = \pi/4$ 25 trajectories are plotted in Fig. 2.

The results for 5000 runs and various α are given in Table 1. Note that average number of reflections equals $N^*/2$.

4.2. Multidimensional case – polyhedral cone Q

For polyhedral cone there exists number M independent of initial data such that billiard trajectory quits Q after no more than M reflections (see [17], also [14], Theorem 7.17). However M depends on geometry of Q . If

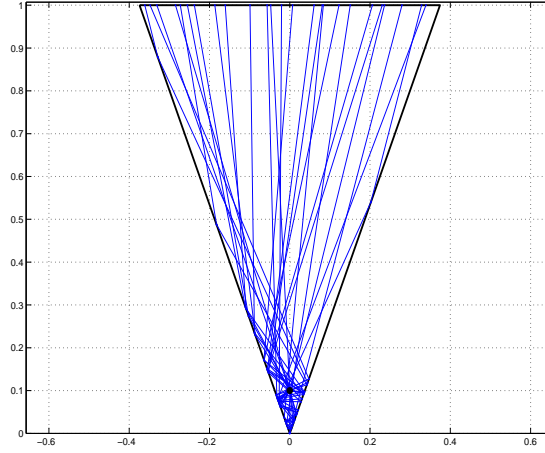


Figure 2: 25 trajectories reaching line $x_2 = 1$ starting from $[0; 0.1]$.

α	Mean	Std
$\pi/2$	2.28	0.87
$\pi/4$	3.08	1.3
$\pi/10$	5.94	2.93
$\pi/50$	25.08	14.46

Table 1: Mean and standard deviation for the number of reflections required to quit the corner with angle α .

M is large ($M > 10n$) then BW algorithm sometimes returns to the initial point. However it is possible to prove, that BW is well defined w.p.1.

4.3. Orthant $Q = \{x \in R^n : x \geq 0\}$

It is easy to note that billiard trajectory independently of initial point and initial direction quits Q after no more than n reflections. Indeed, if d is direction of trajectory, $I = \{i : d_i < 0\}$ then at each reflection I decreases, and after $\leq n$ reflections $I = \emptyset$.

HR trajectory quits Q with probability $1 - (1 - 2^{n-1})$ after a single iteration, thus it requires approximately 2^{n-1} iterations to quit Q with probability $1 - 1/e = 0.63$. Hence BW is much more effective than HR for this case. Simulations for a cube (see below) confirm this statement.

4.4. Convex corner

All these results show that a polyhedral corner is not a problem for BW in contrast with HR, where distance of the initial point to the boundary plays a significant role. The results can be extended for curvilinear corners with nondegenerate linear approximation, i.e. if linear approximation of a corner is a polyhedral cone with nonempty interior.

4.5. Concave corner

It is known [17] that concave corners can be attraction points for billiard trajectories. Consider typical set

$$Q = \{x \in R^2 : \|x\|_\infty \leq 1, \|x - a_i\| \geq 1\} \quad (1)$$

a_i are vertices of $\|x\|_\infty \leq 1$. Then Q has 4 concave corners (cusps). Simulation of the single trajectory starting in $[-0.001; 0]$ in direction $[0; -1]$ is shown in Fig. 3. It requires 121 reflection to get out of the corner. For $x = [-10^{-4}; 0]$ and the same direction it requires 670 reflections, for $x = [-10^{-5}; 0]$ the number of reflections is 3802. But, in general, these “bad” directions are rare. Fig. 4 (left) depicts 200 points for the set (1), the average number of reflections per point is 6.

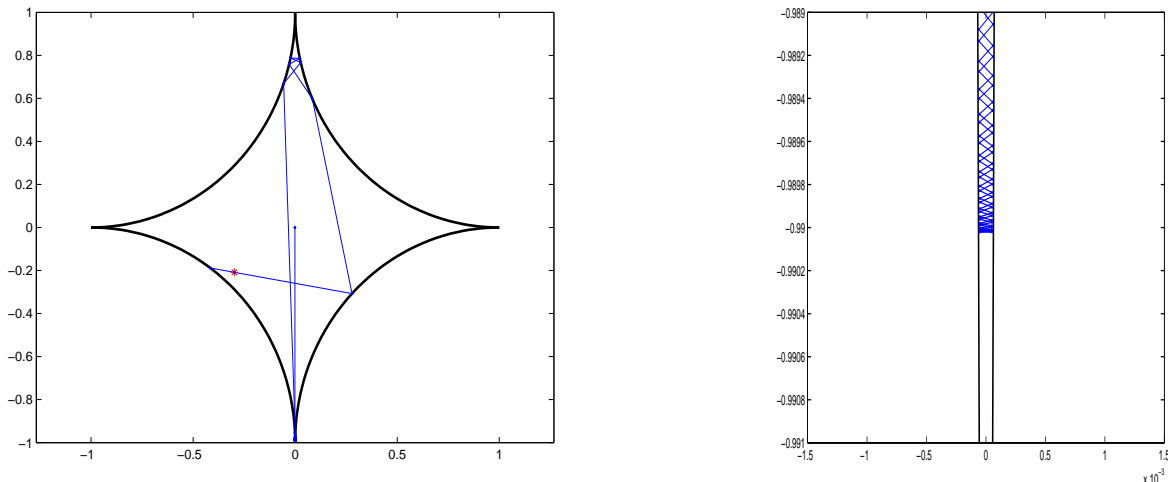


Figure 3: Left: the trajectory for the set (1) starting in $[-0.001; 0]$ in direction $[0; -1]$. Right: zoom of the lower part.

Consider slightly different concave angle

$$Q = \{x \in R^2 : -x_1^4 < x_2 < x_1^4, \quad x_1 \geq 1\} \quad (2)$$

with curvature tending to zero. It may happen that billiard trajectory comes (in limit) to a point with no normal with nonzero probability and for fixed ℓ the length of billiard trajectory remains $< \ell$ after any number of reflections. Indeed, start a trajectory at point $x^0 = [0.9; \varepsilon]$, ε being small enough, fix $\ell = 1$, $d = [-1; 0]$ and calculate the number of reflections needed to realize the trajectory. The results are shown on Table 2.

ε	Number of reflections
1e-3	746
5e-4	1851
4e-4	2480
3e-4	3617
2e-4	6158
1.1e-4	13496
1.01e-4	>5e+6

Table 2: Number of reflection required to realize the trajectory of length 1 for domain (2) starting from $x^0 = [0.9; \varepsilon]$ in direction $d = [-1; 0]$.

As one can notice the number of reflections increases dramatically as the first coordinate of x^0 tends to zero and even for $x_1^0 = 10^{-4}$ the trajectory becomes unreliable. So to be on the safe side of situations like this we restrict the number of reflections in the algorithm. In Fig. 4 (right) 200 samples for domain (2) are depicted, average number of reflections is 5.2.

4.6. Strip

For a set like $Q = \{x \in R^2 : 0 \leq x_2 \leq 1, |x_1| < M\}$, M being large enough HR and BW have different abilities to walk along x_1 . If we count average step per 1 boundary oracle application, BW is approximately 4 times faster. Indeed, HR takes 2 BO for 1 step and takes approximately the middle of a segment, while BW for 1 BO takes the segment height and its shift along x_1 appears to be twice larger than for HR.

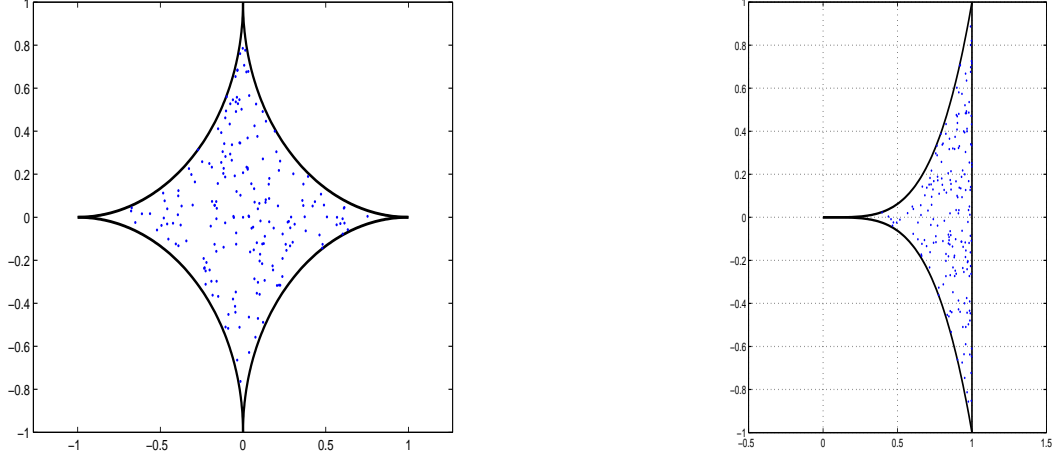


Figure 4: Left: 200 point for domain (1). Right: 200 point for domain (2).

4.7. Cube

For the unit cube $Q = \{x \in \mathbb{R}^n : 0 \leq x \leq 1\}$ (inequality understood component-wise) we can derive the next point of the BW algorithm explicitly.

At the current point x for given ℓ and d calculate $k_i = \lfloor x_i + \ell d_i \rfloor$ ($\lfloor x \rfloor$ is maximal integer less than or equal to x) and walk to y :

$$y_i = \begin{cases} x_i + \ell d_i - k_i, & k_i \text{ is even} \\ 1 - (x_i + \ell d_i - k_i), & k_i \text{ is odd} \end{cases}, \quad i = 1, \dots, n.$$

Of course there is no need to apply MCMC algorithms for random sampling in a cube: a code `rand(n,1)` in MatLab generates uniformly distributed points. Moreover, the shape of a cube is so nice that distribution of HR points converges to uniform fast enough. Nevertheless serial correlation for these points is evident. In Fig. 5 we compare $s_k = E\|x^k - x^0\|_\infty$ for $n = 50$ averaged over 500 runs for two initial points $x^0 = [1/2, \dots, 1/2]^T$ (left) and $x^0 = [1/n, \dots, 1/n]^T$ (right). Implementing BW we take $\tau = \sqrt{n}$. One can see that serial correlation is much stronger for HR samples (black) than for BW samples (blue), and even 100 iterations are not enough for HR algorithm to quit the corner.

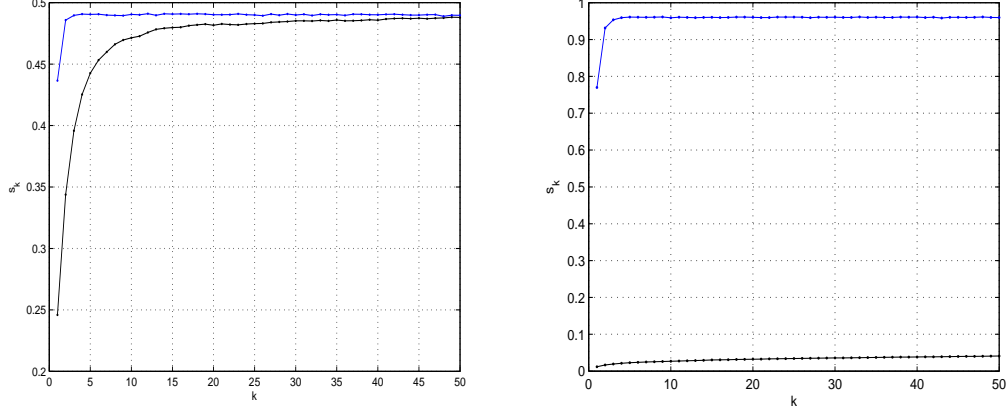


Figure 5: $n = 50$. $E||x^k - x^0||_\infty$ averaged over 500 runs for $x^0 = [1/2, \dots, 1/2]^T$ (left) and $x^0 = [1/n, \dots, 1/n]^T$ (right) for HR samples (black) and BW samples (blue).

4.8. Simplex

The next test set is standard n -dimensional simplex

$$Q = \{x_i \geq 0, \sum x_i = 1, i = 0, 1, \dots, n\}.$$

Simplex is a set, containing many corners, while the geometry of simplex can't be improved by any affine transformation. We know that for HR walk it takes a lot of iterations to get out of a corner, thus it is interesting to compare HR and BW.

Smooth boundary of Q is specified by points $B = \{x \in \mathbb{R}^{n+1} : x_k = 0 \text{ for one } k\}$ and internal normal of unit length at such points is

$$s = \sqrt{\frac{1}{n(n+1)}} \left[-1, \dots, \underbrace{n}_k, \dots, -1 \right]^T.$$

Edge length is $\sqrt{2}$ for every dimension n and we take $\tau = \sqrt{2}$. Fig. 6 shows 300 points generated by HR (black) and BW (blue) for standard 2-simplex.

We see that for $n = 2$ samples look uniformly distributed for both algorithms. To judge about the uniformity more rigorously in multidimensional case, consider the sequence of enclosed simplices $S_\alpha = \{x \in \mathbb{R}^{n+1} : x_i \geq \alpha, \sum x_i = 1\}$, $0 \leq \alpha \leq \frac{1}{n+1}$. For $\alpha = 0$ we have the initial simplex, for

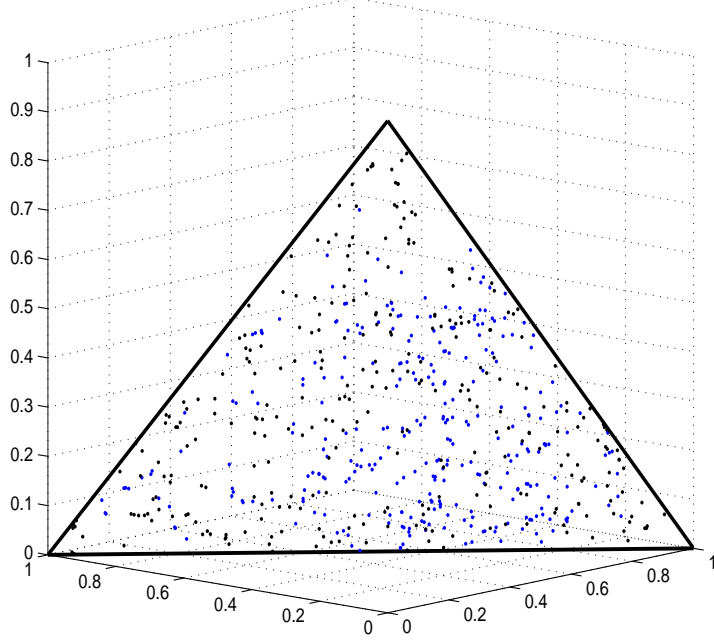


Figure 6: 300 points generated by HR (black) and BW (blue) points for standard 2-simplex.

$\alpha = \frac{1}{n+1}$ simplex S_α contains one point. Let $\hat{f}(\alpha)$ be a portion of points contained in S_α , and denote $f(\alpha) = \text{vol}S_\alpha / \text{vol}S_0 = (1 - (n+1)\alpha)^n$. Fig. 7 shows $\hat{f}(\alpha)$ for $n = 50, N = 300, x^0 = \{1/(n+1), \dots, 1/(n+1)\}$. Red line corresponds to uniformly distributed points, black line describes the distribution for HR points and blue line for BW points. We conclude that for BW samples empirical values of $\hat{f}(\alpha)$ are much closer to mean value $f(\alpha)$ than for HR samples.

Another advantage of BW — its ability to quit a corner fast enough. We take a specific starting point close to the corner $x^0 = \left[1 - \varepsilon, \frac{\varepsilon}{n}, \dots, \frac{\varepsilon}{n}\right]^T$, $\varepsilon = 0.1$.

Fig. 8 depicts the distance between x^i and the vertex of the simplex $c = [1, 0, \dots, 0]^T$. HR is unable to get out of the corner while the behavior of BW points after a few iterations looks like for uniformly distributed points.

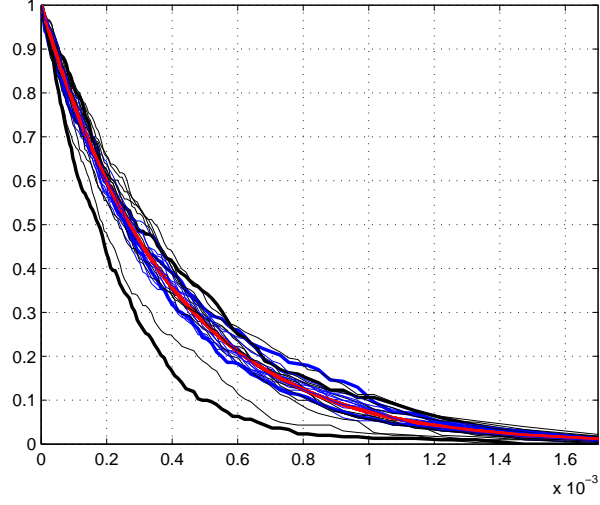


Figure 7: Portion of points contained in S_α for uniformly distributed points (red), HR (black) and BW (blue). $n = 50, 300$ points.

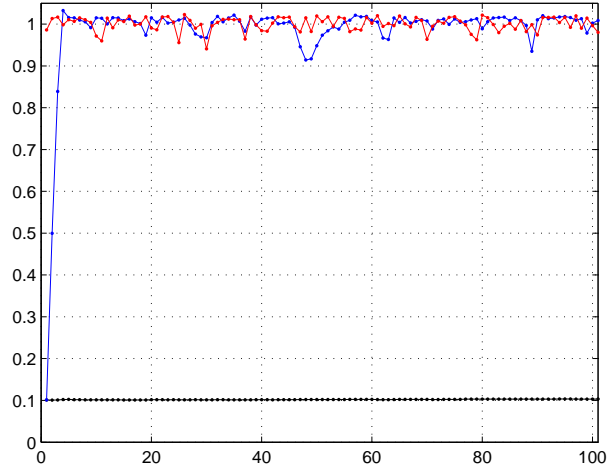


Figure 8: Distance from the corner $[1, 0, \dots, 0]^T$ for the first 100 points of walks in 50-simplex for different starting points. $x^0 = \left[1 - \varepsilon, \frac{\varepsilon}{n}, \dots, \frac{\varepsilon}{n}\right]^T$, $\varepsilon = 0.1$. Uniformly distributed points (red), HR (black) and BW (blue).

4.9. Toroid

Both algorithms HR and BW are applicable for nonconvex sets. Consider the toroid formed by a n -dimensional ball of radius r with its center rotating in a circle in (x_1, x_2) -plane:

$$Q = \{x \in \mathbb{R}^n : \|x - c_x\| \leq r\}, \quad (3)$$

where $c_{xi} = \frac{x_i}{\sqrt{x_1^2 + x_2^2}}$, $i = 1, 2$, $c_{xi} = 0$, $i > 2$.

Fig. 9 demonstrates $N = 1000$ samples (projected onto (x_1, x_2) -plane) for the set (3) with $r = 1/3$ of dimension 10 (x_1, x_2) -plane. HR points are plotted with black dots, BW points with blue.

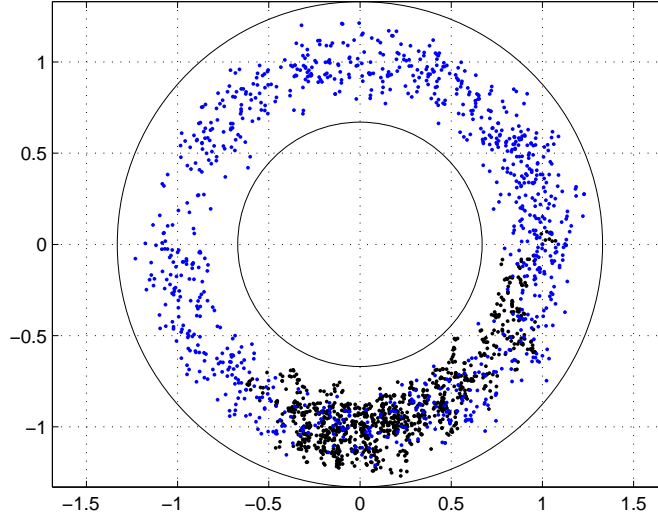


Figure 9: (x_1, x_2) -projection for HR points (black) and BW points (blue) for toroid (3). $n = 10$, $N = 10^3$.

It can be easily seen that angles of BW points are much more uniformly distributed than for HR points, which remain in the neighborhood of the initial point.

5. Applications

In this paper we do not address numerous applications of new version of random sampling. We can mention just few of them — global optimization

(in particular, concave programming), control problems, robustness issues, numerical integration, calculation of volume and of center of gravity and so on, see, for instance, our previous papers [9, 10, 11, 13]. We plan to consider these applications in future works.

6. Acknowledgements

The initial impulse for the research was given by a student Alexander Rodin, who proposed to use physical model of gas diffusion for random sampling. The advices of Yakov Sinai on billiard theory were extremely helpful. The study was supported by Laboratory of Structural Methods of Data Analysis in Predictive Modeling in Moscow Institute of Physics and Technology (“mega-grant” of Russian Government).

References

- [1] R. Tempo, G. Calafiore, F. Dabbene, Randomized Algorithms for Analysis and Control of Uncertain Systems, Communications and Control Engineering Series, Springer-Verlag, London, 2004.
- [2] R. Rubinstein, D. Kroese, Simulation and the Monte Carlo Method, Wiley, NJ, 2008.
- [3] W. Gilks, S. Richardson, D. Spiegelhalter, Markov Chain Monte Carlo, Chapman and Hall, 1996.
- [4] P. Diaconis, The markov chain monte carlo revolution, Bull. of the AMS 46 (2) (2009) 175–205.
- [5] M. Dyer, A. Frieze, R. Kannan, A random polynomial-time algorithm for approximating the volume of convex bodies, Journal of the ACM 38 (1) (1991) 1–17.
- [6] L. Lovasz, M. Somonovits, Random walks in a convex body and an improved volume algorithm, Random Structures & Algorithms 4 (4) (1993) 359–412.
- [7] V. Turchin, On the computation of multidimensional integrals by the monte carlo method, Theory of Probability and its Applications 16 (4) (1971) 720–724.

- [8] R. Smith, Efficient monte carlo procedures for generating points uniformly distributed over bounded regions, *Operations Research* 32 (6) (1984) 1296–1308.
- [9] B. Polyak, E. Gryazina, Hit-and-Run: New design technique for stabilization, robustness and optimization of linear systems, in: *Proceedings of the IFAC World Congress*, 2008, pp. 376–380.
- [10] B. Polyak, E. Gryazina, Randomized methods based on new Monte Carlo schemes for control and optimization, *Annals of Operational Research* 189 (1) (2011) 343–356.
- [11] F. Dabbene, P. Shcherbakov, B. Polyak, A randomized cutting plane method with probabilistic geometric convergence, *SIAM Journal of Optimization* 20 (6) (2010) 3185–3207.
- [12] Y. Nesterov, A. Nemirovsky, *Interior Point Polynomial Methods in Convex Programming*, SIAM, Philadelphia, 1994.
- [13] B. Polyak, E. Gryazina, Markov Chain Monte Carlo method exploiting barrier functions with applications to control and optimization, in: *IEEE Multi-Conference on Systems and Control*, 2010, pp. 1553–1557.
- [14] S. Tabachnikov, *Geometry and Billiards*, RI: Amer. Math. Soc., 1995.
- [15] G. Galperin, A. Zemlyakov, *Mathematical Billiards*, Nauka, Moscow (in Russian), 1990.
- [16] Y. G. Sinai, Dynamical systems with elastic reflections, *Russian Mathematical Surveys* 25 (2) (1970) 137–189.
- [17] Y. G. Sinai, Billiard trajectories in a polyhedral angle, *Russian Mathematical Surveys* 33 (1) (1978) 219–220.
- [18] V. V. Kozlov, D. V. Treshchev, *Billiards: A Genetic Introduction to the Dynamics of Systems with Impacts*, Vol. 89, *Translations of Mathematical Monographs by American Mathematical Society*, Providence, RI, 1991.
- [19] L. Lovasz, S. Vempala, Hit-and-run from a corner, in: *Proceedings of the 36th annual ACM symposium on Theory of computing*, Chicago, IL, USA, 2004, pp. 310–314.